

In the claims

1. (Currently Amended) A method for modifying a computer application in substantially real-time without suspending or terminating the application, said method comprising the steps of:
 - (a) connecting to an application executing on an application server, the application server having a computer memory;
 - (b) acquiring program data ~~related to~~ that defines the underlying program structure including one or more object language components of the application;
 - (c) displaying the program data including the one or more object language components to a maintenance person;
 - (d) accepting a command from the maintenance person; and
 - (e) executing the command to cause the program data of the executing application to be modified without suspending or terminating the executing application.
2. (Original) The method of claim 1, further comprising the step of modifying data that has been cached in the computer memory.
3. (Original) The method recited in claim 1, further comprising the step of modifying an order of the execution of a plurality of methods within the application.
4. (Original) The method recited in claim 1, further comprising the step of providing more detailed diagnostic messages in response to the command.
5. (Currently Amended) The method recited in claim 1, further comprising the step of connection to the application using ~~Java~~ JAVA programming language RMI.
6. (Original) The method recited in claim 1, further comprising the step of modifying an application written in an interpreter programming language.
7. (Original) The method recited in claim 1, further comprising the steps of:
 - (a) accepting a selection of a method from the program data; and

(b) invoking the method from the command line with at least one new argument.

8. (Original) The method recited in claim 1, further comprising the steps of:

- (a) accepting a selection of a method from the program data;
 - (b) prompting the maintenance person for at least one new argument value;
- and
- (c) invoking the method with the at least one argument.

9. (Currently Amended) A system for modifying an application in substantially real-time during execution without suspending or terminating the application, comprising:

an application server on which the application executes;
an object shell console that attaches to the application while it is running to obtain program data defining the underlying program structure of the application including at least one object language component;

a graphical user interface in the object shell console that is used to assist a maintenance person in modifying the program data of the application; and

a command line for accepting a command to be executed, said command when executed will cause the execution of the application to be modified without suspending or terminating the application.

10. (Original) The system recited in claim 9, further comprising:

a vector for establishing an order of method execution in the application;
a command line for entering a new vector comprising a different order for executing the methods in the application; and

wherein entering the new vector in the command line establishes the different order of method execution.

11. (Original) The system recited in claim 9, further comprising a data cache in a memory of the application server, said data cache being modified by the command.

12. (Original) The system recited in claim 9, further comprising:
a terse logging operation that provides terse diagnostic messages;
a detailed logging operation that provides detailed diagnostic messages; and
means for allowing a maintenance person to select detailed diagnostic messages if
an error occurs without suspending or terminating the application.
13. (Currently Amended) The system recited in claim 9, wherein the object shell
console attaches to the application using Java JAVA programming language RMI.
14. (Original) The system recited in claim 9, wherein the application is written in an
interpreter programming language.
15. (Original) The system recited in claim 9, wherein the program data comprises at
least one method related that is executed in the application, further comprising:
means for accepting a selection of a method from the program data; and
means for invoking the method from the command line with at least one new
argument.
16. (Original) The system recited in claim 9, wherein the program data comprises at
least one method related that is executed in the application, further comprising:
means for accepting a selection of a method from the program data;
a prompt to prompt the maintenance person for at least one new argument value;
and
means for invoking the method with the at least one new argument value.
17. (Currently Amended) A system for modifying an executing application in
substantially real-time, comprising:
a computer on which the application is executing;
means for attaching to the executing application so that program data that defines
the underlying program structure and that includes at least one object language
component is extracted from the executing application;

a display device for displaying the program data to a maintenance person;
means for accepting a command from the maintenance person; and
means for invoking the command to thereby cause the program data of the
application to be modified in accordance with the command without suspending or
terminating the application.

18. (Original) The system recited in claim 17, wherein the application is written in an
interpreter programming language.

19. (Currently Amended) The system recited in claim 17, wherein the attaching
means further comprises means for attaching to the application using ~~Java~~ JAVA
programming language RMI.

20. (Original) The system recited in claim 17, wherein the program data includes one
or more methods that are executing in the application, further comprising:
means for accepting a selection of a method from the display of the program data;
and
means for invoking the method from the command line with at least one new
argument.

21. (Original) The system recited in claim 17, wherein the program data includes one
or more methods that are executing in the application, further comprising:
means for accepting a selection of a method from the display of the program data;
a prompt to prompt the maintenance person for at least one argument value; and
means for invoking the method with the at least one argument.